Adaptive Ensemble Learning for Object Counting under varying Environmental Conditions

Szabolcs Péter Gombos
gombosszabi01@gmail.com
djalet@inf.elte.hu
ELTE Eötvös Loránd University
Budapest, Hungary

László Gulyás
lgulyas@inf.elte.hu
ELTE Eötvös Loránd University
Budapest, Hungary

Abstract

Object counting, such as counting people in frames with varying crowd densities, remains challenging due to diverse and dynamic environmental conditions within or across image sequences. Traditional counting models typically specialize in either dense or sparse crowd scenes, limiting their effectiveness for images with mixed densities, where one region might be dense while another sparse. In this paper, an adaptive ensemble learning framework is introduced that is capable of context awareness to enable reliable object counting across varying environmental conditions. The method dynamically combines multiple specialized models optimized for different density scenarios, leveraging their individual strengths based on the spatial density distribution within each frame. Using an adaptive weighting mechanism, this framework selects the optimal combination of predictions from specialized models (for example: Faster R-CNN/Yolo for sparse scenarios and CSRNet for dense scenarios), ensuring accurate counting in complex scenes. The resulting gain in counting accuracy is especially valuable for autonomous aerial robots, which depend on dependable crowd-size estimates to support surveillance, search-and-rescue, and other safety-critical missions. Experimental results demonstrate that this adaptive ensemble learning approach significantly enhances counting accuracy compared to single-model solutions, effectively handling heterogeneous crowd distributions and environmental variability.

Keywords

Object counting, People counting, Ensemble learning, Adaptive ensemble learning

ACM Reference Format:
Szabolcs Péter Gombos and László Gulyás. 2025. Adaptive Ensemble Learning for Object Counting under varying Environmental Conditions. In *Proceedings of Intelligent Robotics FAIR 2025 (IntRob'25).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
IntRob'25, Budapest, HU
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Object counting from images, such as counting people in crowd scenes, has become an increasingly important task in various applications, including surveillance, crowd management, traffic monitoring and public safety [14, 28]. Although remarkable advances have been made in deep learning-based methods for image classification, object detection and segmentation [5, 27], reliably counting objects under dynamically changing and heterogeneous environmental conditions remains challenging. This is due to the variability in crowd densities, scale differences, occlusions and illumination changes that are common within and across different image sequences [18].

Traditional crowd-counting approaches typically specialize in either dense or sparse crowd scenes. For dense crowd scenarios, regression-based methods such as CSRNet have proven effective [18] by learning density maps to estimate object counts, especially where objects are densely packed. Conversely, detection-based methods, like Faster R-CNN or YOLO, show high accuracy for sparse scenes by detecting and counting individual objects. However, applying a single-model approach limits effectiveness when the scene presents a mixed density scenario, such as images with densely populated regions alongside sparse areas.

Adaptive ensemble learning provides a promising direction to address these limitations. Ensemble methods are known for combining multiple specialized models to leverage their individual strengths, thus enabling robust predictions even under varied conditions [30]. The key advantage of ensemble learning lies in its flexibility to dynamically select and combine model predictions based on contextual awareness, effectively adjusting to spatial density variations within the same image. This approach can be particularly beneficial for robotic applications, such as autonomous aerial vehicles (drones) tasked with estimating crowd sizes to inform decision-making in surveillance, disaster response, or public safety missions. Accurate and robust object counting capabilities in drones directly contribute to more effective deployment and operational efficiency, especially when managing dynamically changing environments [19, 25].

This paper proposes an adaptive ensemble learning framework specifically designed for object counting under varying environmental conditions. The approach uses DeepLabv3 [5] to partition the original image into meaningful regions corresponding to potential object clusters. The segmented regions (or slices) are classified into dense and sparse crowd scenarios by a fine-tuned model (based on ResNet-50). Densely populated regions are processed by CSRNet [18], while Faster R-CNN [28] is used for sparse region.

The proposed method is validated on the JHU-Crowd [11] and the MALL [1] datasets which are published on Kaggle. Experimental results demonstrate that the adaptive ensemble framework

significantly enhances counting accuracy compared to traditional single-model approaches, effectively handling heterogeneous crowd distributions and adapting robustly to environmental variability.

The remainder of this paper is organized as follows. Section 2 reviews related works. This is followed by the details of the proposed solution in Section 3. Section 4 discusses the experimental settings and datasets used. The findings are discussed in Section 5. Section 6 concludes the paper.

2 Related Works and Background

2.1 Classic object counting methods

Early object counting pipelines relied on simple, hand-crafted steps. First, background subtraction (for example, frame differencing or Gaussian mixture models) isolated moving pixels [32], a series of morphological filters (erosion, dilation, opening/closing) then cleaned noise and produced contiguous foreground blobs [31]. Alternatively, contour-based methods (Canny edges [2], Hough transforms [10]) traced object outlines directly from gradients or thresholding. A third strategy, motion-based detection, used optical flow [15, 22] or frame-to-frame differencing (often paired with Kalman or particle filters) to cluster coherent motion into tracked entities. These algorithms often required scene-specific calibration of thresholds and kernel sizes, making deployment laborious. Many implementations attempted shadow removal heuristics, yet even sophisticated shadow cancelation fails under sudden lighting changes. Their reliance on fixed morphological parameters further complicates adaptation to objects of varying scale. Although these methods are computationally inexpensive, they are highly sensitive to illumination changes, dynamic backgrounds and cast shadows. Moreover, their accuracy degrades drastically in crowded scenes, where many objects overlap or occlude one another. This makes them unsuitable for dense-crowd scenarios.

2.2 Deep learning methods in object counting

The advent of deep learning has ushered in a paradigm shift in object counting. By training large convolutional networks end-to-end on rich, annotated imagery, these methods learn hierarchical feature representations,ranging from low-level edges and textures up to high-level object concepts, without manual tuning of filters or thresholds. As a result, deep models exhibit far greater robustness to variations in scale, viewpoint, illumination and background clutter than classical pipelines. These methods broadly fall into two main categories, which are Detection-based[27, 28, 35] and Regression/Density-based[18].

2.2.1 Regression-based methods. Regression-based models bypass explicit detection by learning to predict a per-pixel density map whose integral over the image yields the total object count. Early architectures such as MCNN [35] addressed scale variation by using multiple parallel convolutional columns with different receptive fields, while SANet [3] introduced a scale aggregation network to fuse multi-scale features. CAN [33] further incorporated context aggregation modules to capture long-range dependencies and Bayesian-loss frameworks [24] replaced the standard L_2 loss with a probabilistic formulation to better handle annotation noise.

Among these, CSRNet [18] has become a standard reference due to its simple yet powerful use of dilated convolutions to preserve spatial resolution while enlarging the receptive field. In the present work, the publicly released CSRNet weights are loaded prior to evaluation to leverage the knowledge distilled on large-scale crowd datasets [23].

CSRNet builds on the first 13 convolutional layers of VGG-16 as its "frontend", loading ImageNet-pretrained weights into those layers. It then appends a "backend" of six dilated convolutional layers (channel sizes [512, 512, 512, 256, 128, 64] and uniform dilation rate = 2) and a final 1×1 convolution to produce the single-channel density map. This design expands the network's receptive field without any further pooling, preserving spatial resolution throughout. This design effectively captures the complex spatial patterns present in highly congested scenes and produces smooth density estimates, achieving state-of-the-art performance on benchmarks with very high crowd densities.

Despite its success, CSRNet exhibits certain limitations. The fixed Gaussian kernel assumes uniform annotation noise and may misrepresent scale variation across different scenes. Furthermore, the purely convolutional architecture lacks explicit attention mechanisms, occasionally producing blurred or underestimated density peaks in regions with extreme scale heterogeneity. Importantly, regression-based models are not well suited to counting very small groups or isolated individuals, as their smoothing operations tend to obscure single-object peaks and can undercount in low-density areas. These observations have inspired follow-up works that integrate scale-aware modules, spatial attention or multi-column feature fusion to further improve robustness under diverse environmental conditions.

2.2.2 Detection-based methods (Faster R-CNN, YOLO). In low density or sparse settings, where individual objects are well separated, detection-based techniques excel by locating each instance explicitly. Detection-based methods frame counting as a by-product of instance detection. Two-stage networks such as Faster R-CNN [28] first employ a Region Proposal Network (RPN) [28] to generate candidate object regions, then refine each proposal via separate classification and bounding-box regression heads.

Faster R-CNN[1] combines a ResNet-50 backbone with a Feature Pyramid Network (FPN) neck [20]. It is initialized with weights pretrained on COCO 2017 [21] [8]. The ResNet-50 backbone begins with a 7×7 convolution and max-pooling layer, followed by four residual stages (conv2_x through conv5_x) that produce feature maps at strides of 4, 8, 16 and 32. The FPN adds lateral 1×1 convolutions from each stage, upsamples higher-level feature maps and merges them via elementwise summation to build a pyramid of multi-scale feature maps (P2–P6). The RPN slides over these feature maps to propose anchors at multiple scales and aspect ratios, generating Regions of Interest (RoIs) [7, 13] that are then pooled via RoIAlign and fed into two fully connected layers for per-RoI classification and bounding-box regression. At inference, predicted boxes with class scores above a confidence threshold (≥ 0.5) are retained and non-maximum suppression (IoU ≥ 0.5) removes duplicate detections.

[1]This work uses the `fasterrcnn_resnet50_fpn` implementation from Torchvision [8].

One-stage detectors such as YOLO ("You Only Look Once") [27] collapse proposal and classification into a single pass, achieving real-time inference by dividing the image into a grid and directly regressing box coordinates and confidence scores. While one-stage models offer dramatic speedups, they generally sacrifice some precision compared with two-stage approaches and may struggle with small or heavily occluded objects.

Overall, detection-based pipelines excel at counting well-separated instances, but their performance degrades in dense crowds and under severe occlusion.

2.3 Semantic segmentation with DeepLabv3

Semantic segmentation provides a pixel-level understanding of the scene, enabling the counting system to distinguish between foreground objects and irrelevant background regions before any instance or density estimation is performed. DeepLabv3 [5, 6] is a particularly effective choice: it augments a standard convolutional backbone (for example, ResNet or Xception) with two key components that preserve spatial resolution while capturing multi-scale context.

First, DeepLabv3 replaces the final pooling layers with a cascade of *atrous* (dilated) convolutions at different dilation rates. In doing so, it expands the receptive field of the network, allowing it to incorporate information from both local details and distant pixels without reducing the resolution of feature maps. Second, the *Atrous Spatial Pyramid Pooling* (ASPP) [6] module applies several parallel atrous convolutions (typically with dilation rates 6, 12 and 18) alongside global average pooling. The outputs are concatenated and refined via a 1×1 convolution. This design jointly captures fine edges, medium-scale structures and global context, which is critical for delineating object boundaries in cluttered or crowded scenes.

In practice, DeepLabv3 produces high-quality masks of target classes (for example, pedestrians, vehicles) even under challenging conditions such as partial occlusion, low contrast or complex backgrounds. These masks serve one very important purpose in a counting system: they restrict subsequent counting networks to regions likely to contain objects. This reduces false positives on background clutter. It also supports adaptive processing by indicating where dense or sparse counting models should be applied.

3 The Proposed Solution

The proposed counting method proceeds in five stages (see Figure 1). First, DeepLabv3 is applied to the input image to produce a high resolution binary foreground mask of all object bearing pixels. Next, a connected components analysis groups these foreground pixels into contiguous regions (see Figure 2 for the effect) and only those regions that exceed a minimum area (for example, 50×50 px) are retained as candidate "slices."[2] Each slice is then resized or padded to a fixed input size and passed through a ResNet-50 classifier, fine-tuned specifically for slice-level density classification (see Figure 3).[3] Slices classified as dense regions are routed to CSRNet, while sparse ones are processed by Faster R-CNN. Finally, the counts from all slices are aggregated to form the overall count estimate for the entire image.

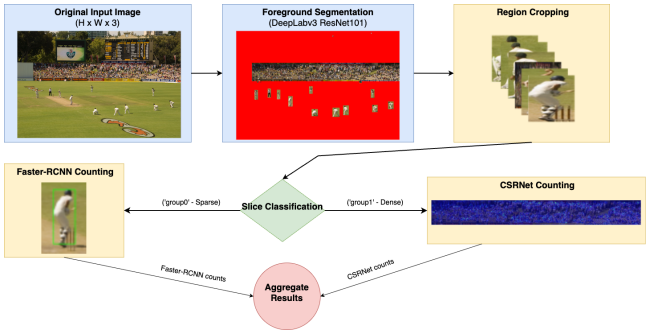

Figure 1: Adaptive ensemble counting pipeline applied to a sports scene.

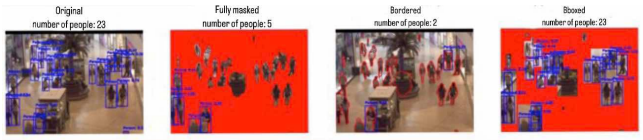

Figure 2: Importance of semantic segmentation. Fully masking the surroundings severely degrades accuracy, whereas enlarging the bounding boxes preserves context and maintains reliable counts.

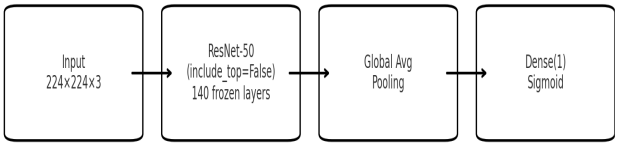

Figure 3: Architecture of the fine-tuned ResNet-50 binary classifier

4 Methods and Data

4.1 Data

Two publicly available datasets were used to train and evaluate the proposed pipeline: the MALL dataset[1] and JHU-Crowd v2.0[11] (see Figure7). JHU-Crowd v2.0 comprises over 4300 images (2200 for training, 500 and 1600 for validation and test, respectively). The MALL dataset contains 2000 frames captured in a shopping mall corridor under various crowd densities. To construct the training

[2]To suppress spurious detections, any slice smaller than 50 × 50 pixels is discarded as noise and not classified.

[3]The fine-tuning of the ResNet-50 based classifier involved the training of the final 52 convolutional layers and the classification head on labeled slices, using binary cross-entropy loss and the Adam optimizer with learning rate 1×10^{-5}, batch size: 32 and early stopping at 3 epochs of no improvement, achieving `val_accuracy` = 96.64%, `val_loss` $\approx$ 0.124.

set for the slice classifier 450 images were sampled from the JHU training set to cover various lighting conditions, camera angles and crowd densities.

For evaluation, a balanced subset of 600 images was assembled by sampling 300 frames from the JHU-Crowd v2.0 test split and 300 from the MALL dataset. These evaluation images underwent the same segmentation and slicing pipeline, followed by slice classification and adaptive counting, providing a robust assessment of generalization across diverse environments and density regimes.

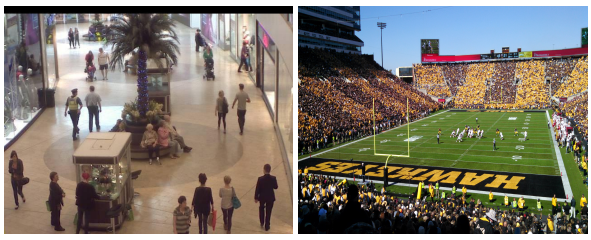

Figure 4: Examples from the MALL (left) and JHU (right) datasets.

4.1.1 Accelerated manual annotation. A special dataset was used to fine-tune the ResNet-50 based region classifier with over 2000 slices extracted from the training datasets. [4] The manual labeling of these slices was prohibitive. Therefore, a semi-supervised strategy was adopted. The raw slice set was first automatically clustered and the resulting clusters were inspected and manually labeled with the following steps:

(1) **Feature extraction:** Slices were resized to a 224×224×3 RGB image and fed through a ResNet-50 pretrained on ImageNet. A 2048-dimensional descriptor was extracted from the global average pooling layer.
(2) **Dimensionality reduction:** Principal Component Analysis (PCA) was applied to map the 2048-dimensional descriptor to 100 principal components (preserving appr. 97% of total variance).
(3) **Standardization:** The 100-dimensional vectors were standardized to zero mean and unit variance.
(4) **Agglomerative clustering:** $n = 2$ clusters with Ward linkage and Euclidean distance were created (resulting in an appr. 60%-40% split).
(5) **Manual correction**: Cluster contents were manually verified. The 60% cluster was mostly retained as "dense", while misclassified examples in the smaller cluster were manually corrected.

4.2 Experimental Setup and Evaluation Metrics

Experiments were carried out to study the relative performance of the proposed solution. It was compared to the performance of both advanced individual counting architectures (CSRNet, YOLO and Faster R-CNN) and to static, non-adaptive ensembles (average, summation, minimum and maximum) of them.

[4]The complete dataset was published on Kaggle. [12]

The various counting approaches are compared using a number of different metrics. **Oracle accuracy** (also known as the "winner-takes-all" metric) [26] quantifies how often each model is the single best predictor on a per-image basis. For each test image, the model whose absolute error is smallest is deemed the "winner". Aggregating these wins over the entire dataset then yields the percentage of cases in which each model would be chosen by an oracle selector. This measure highlights the potential upper-bound performance if one could perfectly choose the best model for each instance [17].

Standard error metrics like **Mean Squared Error (MSE)** and **Root Mean Squared Error (RMSE)** are also applied. Since these can be unduly influenced by a small number of extreme deviations, **Median Absolute Error (MedAE)** and **Mean Absolute Percentage Error (MAPE)** are also used. [4, 16, 29, 34]

Finally, **rank-based scoring** is used to capture the relative ordering of model performance on each test image. Methods are ranked based on the absolute errors produced and scored according to these ranks. This metric emphasizes consistent outperformance across instances and helps balancing differences in error scales.[9]

5 Results

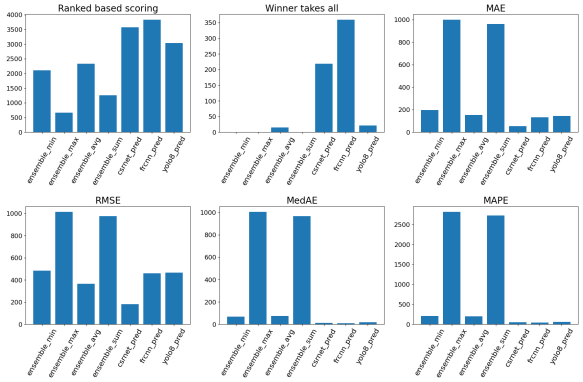

Figure 5: Evaluation of basic ensembles (from left to right: min, max, average, sum) and the three individual models (CSRNet, Fasr R-CNN and YOLOv8). The different panels account for different metrics (left-to-right in the top: rank-based scoring, winner takes all, MAE, in the bottom: RMSE, MedAE and MAPE).

Figure 5 presents six complementary evaluation metrics for the three backbone models (CSRNet, Faster R-CNN, YOLOv8) and four static ensemble schemes (min, max, avg, sum). The panels show, clockwise from top left: rank-based scoring, winner-takes-all accuracy, MAE, MAPE, MedAE and RMSE. Clearly, none of the static ensembles could significantly and consistently improve the performance of the individual base models. While the aggregations based on averaging and summation partially compensated for the errors of the individual models, they could not effectively deal with situations where one of the models deviated significantly from the true values. Minimum and maximum ensembles were even less successful, as they tend to be distorted by the most extreme prediction. These results demonstrate that traditional ensemble techniques

alone are not a solution to deal with mass density diversity and therefore an adaptive gating mechanism is needed.

Model	Closest Cases	Percentage (%)
Adaptive Ensemble	**413**	**68.83**
CSRNet	128	21.33
Faster R-CNN	346	57.67
YOLOv8	20	3.33

Table 1: Comparison of the proposed Adaptive Ensemble to baseline counting architectures. Each row represents an independent evaluation where predictions from each individual model, tested separately, were compared against the ground truth. The "Closest Cases" and "Percentage (%)" columns indicate how often each standalone model provided the closest prediction to the ground truth on a sample of 600 test images.

Table 1 compares the proposed Adaptive Ensemble to the baseline counting models. The number and percentage of test images is reported for which the method's prediction was closest to the ground truth. Clearly, the Adaptive Ensemble decisively outperforms the other methods, winning 413 out of 600 images (**69%**), compared with 346 (**58%**) for Faster R-CNN, 128 (**21%**) for CSRNet and just 20 (**3%**) for YOLOv8.

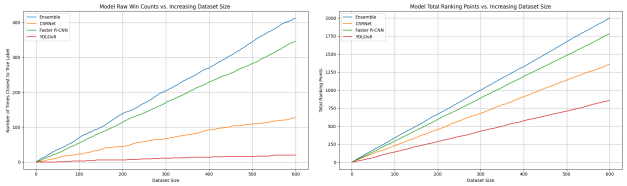

Figure 6: Winner takes performance (left) and rank-based score (right) of the various methods over an increasing test (from 0 to 600 images) set (blue: Adaptive Ensemble, orange: CSRNet, green: Faster R-CNN, red: YOLOv8)

Figure 6 (left) plots the cumulative number of "wins" (cases where a model's prediction is closest to the ground truth) against the size of the test set (growing from 0 to 600 images). The lines correspond to Adaptive Ensemble (blue), CSRNet (orange), Faster R-CNN (green) and YOLOv8 (red), respectively. The slopes of the lines directly reflect the oracle-accuracy rate of the method.

Figure 7 plots MedAE (left panel) as the test set grows from 200 to 600 images, showing the consistently low median error curve (blue) of the Adaptive Ensemble compared to the flatter, higher lines of the individual models. In case of MAPE (right panel) the error of the Adaptive Ensemble (blue) steadily decreases as data is added, while Faster R-CNN (green) and YOLOv8 (red) remain largely unchanged. CSRNet (orange) decreases as well, but flattens out at a much higher error than the Adaptive Ensemble. Table 2 summarises the final performance of the methods on the full test dataset, identifying Adaptive Ensemble as a clear winner, both in terms of MedAE (**9.00** heads) and MAPE (**32.53%**).

Figure 6 (right panel) depicts the cumulative rank-based scores collected as a function of test set size (growing from 0 to 600 images).

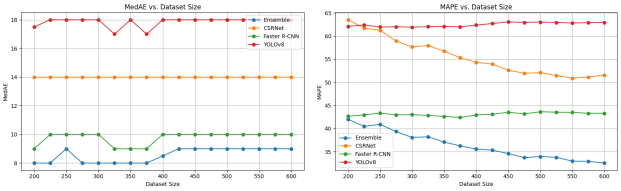

Figure 7: MedAE (left) and MAPE (right) performance of the various models over an increasing test (from 0 to 600 images) set (blue: Adaptive Ensemble, orange: CSRNet, green: Faster R-CNN, red: YOLOv8)

Model	MedAE (heads)	MAPE (%)
Adaptive Ensemble	**9.00**	**32.53**
CSRNet	14.00	51.55
Faster R-CNN	10.00	43.29
YOLOv8	18.00	62.92

Table 2: Median Absolute Error (MedAE) and Mean Absolute Percentage Error (MAPE) on a test set of 600 images

Scores are obtained by assigning $N - rank + 1$ points per image when ranking the methods by absolute error. Adaptive Ensemble (blue) exhibits the steepest slope, reaching 1999 points at 600 images, followed by Faster R-CNN (green, 1783 points), CSRNet (orange, 1362 points) and YOLOv8 (red, 856 points). The context-aware Adaptive Ensemble emerged, again, as the top performer for the full test set of 600 images. It collected roughly **13%** more points than Faster R-CNN and **more than double** the score of YOLOv8 (see Table 3).

Model	Mean points / image	Relative gain vs ensemble
Adaptive ensemble	**3.33**	—
Faster R-CNN	2.91	−13 %
CSRNet	2.25	−32 %
YOLOv8	1.42	−63 %

Table 3: Rank-based scores of the various methods on the 600 images dataset (point/image and relative gain compared to Adaptive Ensemble).

5.1 Running time of the Adaptive Ensemble

Table 4 reports the average processing time of each component of the proposed solution pipeline on a single 720 p (1280 × 720) image: [5] DeepLabv3 segmentation (24 s), connected components and slice extraction (8 s), slice classification (6 s) and counting with CSRNet (18 s) or Faster R-CNN (3 s), resulting in a total runtime of 59 s. Despite the relatively high computational cost, lightweight segmentation alternatives significantly reduced accuracy, thus accuracy remains prioritized here.

[5]Using The the v2-8 TPU configuration of Google Colab (8 TPU cores, each with 45 TOPS and a total of 128 GB of HBM memory).

Processing step	Avg. time
DeepLabv3 segmentation (TPU)	24s
Connected components & slice extraction	8s
Slice classification (avg. 6 slices/image)	6s
CSRNet on dense slices	18s
Faster R-CNN on sparse slices (or black-out image)	3s
Full pipeline	≈ 59 s/image

Table 4: Average inference time per 720 p (1280 × 720) image.

6 Conclusions

This paper presented an adaptive ensemble learning pipeline for object counting under varied environmental conditions. The pipeline combines semantic segmentation (DeepLabv3) with a fine-tuned ResNet-50 classifier, employing an adaptive gating mechanism to direct dense regions to CSRNet and sparse regions to Faster R-CNN. Experiments on two public benchmark datasets (JHU-Crowd, MALL) confirmed the superiority of this approach compared to individual models (CSRNet, Faster R-CNN, YOLOv8) and static ensembles, in terms of MAE and RMSE. Rapid manual annotation significantly accelerated the fine-tuning of the ResNet-50-based region classifier, achieving a validation accuracy of 96.6%. By leveraging off-the-shelf models, the adaptive ensemble not only simplifies implementation and deployment but also delivers superior counting accuracy in heterogeneous environments. However, handling continuously varying densities within segments remains challenging. Future work could incorporate depth-based segmentation to enhance spatial granularity and accuracy, beneficial for aerial robots in surveillance and rescue missions.

References

[1] [n. d.]. Kaggle Crowd Counting Dataset. https://www.kaggle.com/datasets/fmena14/crowd-counting. Accessed: 2024-05-14.

[2] John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 6 (1986), 679–698.

[3] Xiaolong Cao, Zhiqiang Shen, Jianguo Li, and Lequan Yu. 2018. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 734–750.

[4] Tianfeng Chai and Roland R. Draxler. 2014. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)?–Arguments against Avoiding RMSE in the Literature. *Geoscientific Model Development* 7 (2014), 1247–1250. doi:10.5194/gmd-7-1247-2014

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587* (2017). https://doi.org/10.48550/arXiv.1706.05587

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. DeepLabv3+: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 833–851. https://arxiv.org/abs/1802.02611

[7] Deepsense.ai Contributors. 2025. Region of Interest Pooling and Region of Interest Align explained. https://deepsense.ai/blog/region-of-interest-pooling-explained/. Blog post, accessed 2025-04-16.

[8] PyTorch Contributors. 2024. Torchvision Detections: fasterrcnn_resnet50_fpn. https://pytorch.org/vision/stable/models.html#torchvision.models.detection.fasterrcnn_resnet50_fpn. Version 0.15.0.

[9] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7 (2006), 1–30.

[10] Richard O. Duda and Peter E. Hart. 1972. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM* 15, 1 (1972), 11–15.

[11] Hiba Fathi. 2023. JHU-Crowd v2.0 Dataset. https://www.kaggle.com/datasets/hibafathi/jhu-crowd-v2-0. Accessed: 2025-05-19.

[12] Szabolcs Peter Gombos. 2025. Dense vs. Sparse Slices Dataset. https://www.kaggle.com/datasets/84a20acf44d1165fcb7817797f592b6b888653feac9505c93bea54f4d1990322. Accessed: 2025-05-20.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2961–2969.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 770–778. https://doi.org/10.1109/CVPR.2016.90

[15] Berthold K. P. Horn and Brian G. Schunck. 1981. Determining Optical Flow. *Artificial Intelligence* 17, 1-3 (1981), 185–203.

[16] Rob J. Hyndman and Anne B. Koehler. 2006. Another Look at Measures of Forecast Accuracy. *International Journal of Forecasting* 22, 1 (2006), 679–688. doi:10.1016/j.ijforecast.2006.03.001

[17] Ludmila I. Kuncheva. 2004. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience.

[18] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 1091–1100. https://doi.org/10.1109/CVPR.2018.00120

[19] Joon Kyu Lim, Seongho Kim, Hyunwoo Kim, and Jaehoon Kim. 2021. Drone-Based Crowd Density Estimation and Localization Using Temporal and Location Sensitive Fused Attention Model on Pyramid Features. *Journal of Intelligent Fuzzy Systems* 40, 2 (2021), 3239–3251. https://jisem-journal.com/index.php/journal/article/download/6969/3230/11638

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 936–944. doi:10.1109/CVPR.2017.106

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. https://arxiv.org/abs/1405.0312. arXiv preprint arXiv:1405.0312.

[22] Bruce D. Lucas and Takeo Kanade. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*. 674–679.

[23] Thomas Lucchetta. 2024. How to Build a Crowd Counting Model Using CSRNet. https://www.rootstrap.com/blog/how-to-build-a-crowd-counting-model-using-csrnet Accessed: 2024-02-15.

[24] Cheng Ma, Wei Li, Xiaodi Wang, and Hao Li. 2019. Bayesian Loss for Crowd Count Estimation with Point-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6142–6151.

[25] Naser Hossein Motlagh, Miloud Bagaa, and Tarik Taleb. 2017. UAV-Based IoT Platform: A Crowd Surveillance Use Case. *IEEE Communications Magazine* 55, 2 (2017), 128–134. doi:10.1109/MCOM.2017.1600587CM

[26] Matteo Re and Giorgio Valentini. 2012. *Ensemble Methods: A Review*. CRC Press, Taylor and Francis Group, 563–594.

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 779–788. https://doi.org/10.1109/CVPR.2016.91

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv preprint arXiv:1506.01497* (2015). https://arxiv.org/abs/1506.01497

[29] Peter J. Rousseeuw and Annick M. Leroy. 2005. *Robust Regression and Outlier Detection*. Wiley-Interscience.

[30] Ofer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249. doi:10.1002/widm.1249

[31] Jean Serra. 1982. *Image Analysis and Mathematical Morphology*. Academic Press.

[32] Chris Stauffer and W. Eric L. Grimson. 1999. Adaptive Background Mixture Models for Real-Time Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 246–252.

[33] Yue Wang, Yuan Yuan, and Gang Zeng. 2019. Learning from Synthetic Data for Crowd Counting in the Wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 8198–8207.

[34] Cort J. Willmott and Kenji Matsuura. 2005. Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Climate Research* 30, 1 (2005), 79–82. doi:10.3354/cr030079

[35] Yuhong Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 589–597.